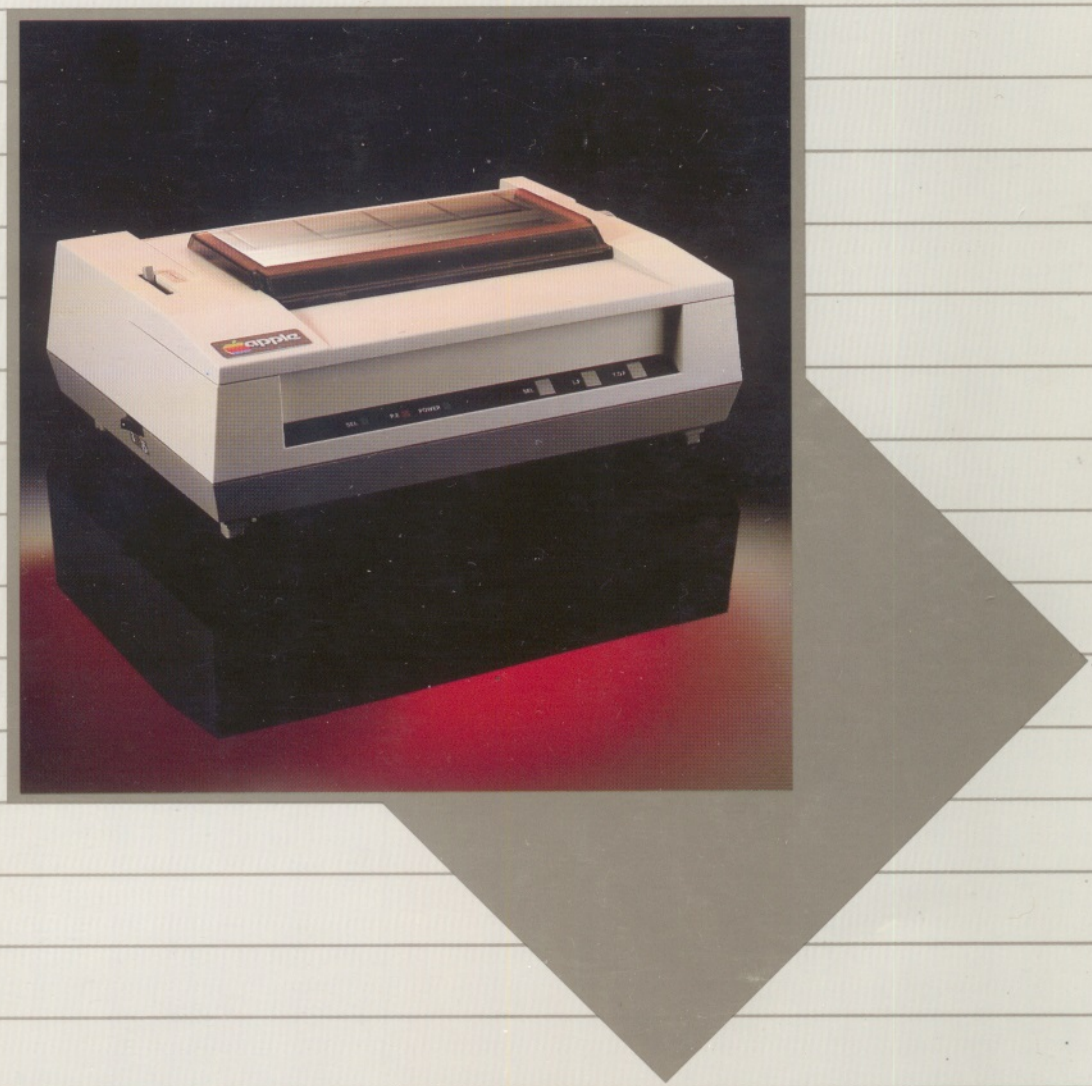
Apple *II*

Dot Matrix Printer User's Manual

Part II: Guide to Apple *II*



Customer Satisfaction

If you discover physical defects in the manuals distributed with an Apple product or in the media on which a software product is distributed, Apple will replace the documentation or media at no charge to you during the 90-day period after you purchased the product.

In addition, if Apple releases a corrective update to a software product during the 90-day period after you purchased the software, Apple will replace the applicable diskettes and documentation with the revised version at no charge to you during the six months after the date of purchase.

In some countries the replacement period may be different; check with your authorized Apple dealer. Return any item to be replaced with proof of purchase to Apple or an authorized Apple dealer.

Limitation on Warranties and Liability

Even though Apple has tested the software described in this manual and reviewed its contents, neither Apple nor its software suppliers make any warranty or representation, either express or implied, with respect to this manual or to the software described in this manual, their quality, performance, merchantability, or fitness for any particular purpose. As a result, this software and manual are sold "as is," and you the purchaser are assuming the entire risk as to their quality and performance. In no event will Apple or its software suppliers be liable for direct, indirect, incidental, or consequential damages resulting from any defect in the software or manual, even if they have been advised of the possibility of such damages. In particular, they shall have no liability for any programs or data stored in or used with Apple products, including the costs of recovering or reproducing these programs or data. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Copyright

Product Revisions

Apple cannot guarantee that you will receive notice of a revision to the software described in this manual, even if you have returned a registration card received with the product. You should periodically check with your authorized Apple Dealer.

© 1983 Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, California 95014

Reorder Apple Product #A2L0079

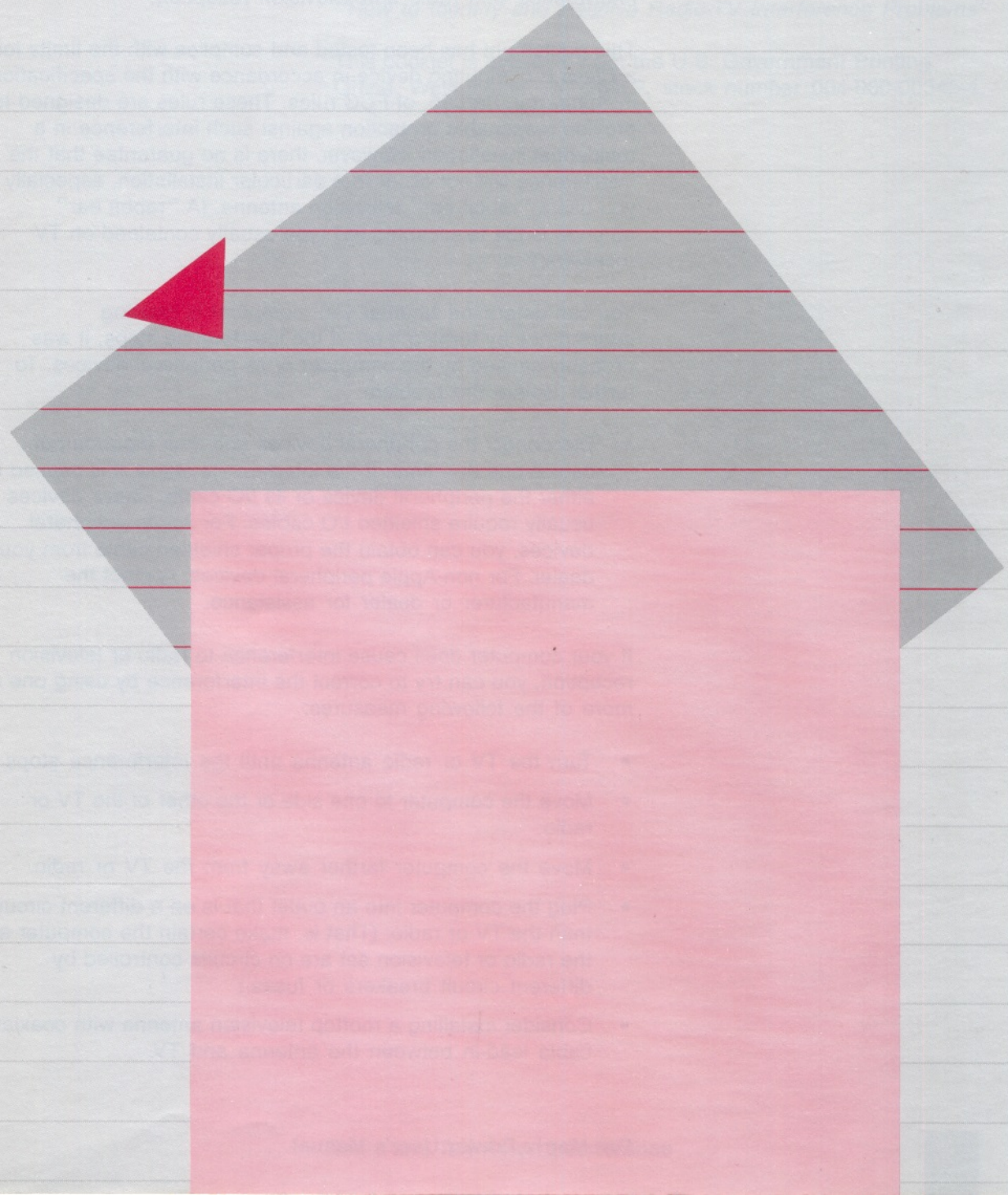
Warning

This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC Rules. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with the Class B limits may be attached to this computer. Operation with non-certified peripherals is likely to result in interference to radio and TV reception.

Apple //

Dot Matrix Printer User's Manual

Part II: Guide to Apple //



Radio and Television Interference

The equipment described in this manual generates and uses radio-frequency energy. If it is not installed and used properly, that is, in strict accordance with our instructions, it may cause interference with radio and television reception.

This equipment has been tested and complies with the limits for a Class B computing device in accordance with the specifications in Subpart J, Part 15, of FCC rules. These rules are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that the interference will not occur in a particular installation, especially if you use a "rabbit ear" television antenna. (A "rabbit ear" antenna is the telescoping-rod type usually contained on TV receivers.)

You can determine whether your computer is causing interference by turning it off. If the interference stops, it was probably caused by the computer or its peripheral devices. To further isolate the problem:

- Disconnect the peripheral devices and their input/output cables one at a time. If the interference stops, it is caused by either the peripheral device or its I/O cable. These devices usually require shielded I/O cables. For Apple peripheral devices, you can obtain the proper shielded cable from your dealer. For non-Apple peripheral devices, contact the manufacturer or dealer for assistance.

If your computer does cause interference to radio or television reception, you can try to correct the interference by using one or more of the following measures:

- Turn the TV or radio antenna until the interference stops.
- Move the computer to one side or the other of the TV or radio.
- Move the computer farther away from the TV or radio.
- Plug the computer into an outlet that is on a different circuit than the TV or radio. (That is, make certain the computer and the radio or television set are on circuits controlled by different circuit breakers or fuses.)
- Consider installing a rooftop television antenna with coaxial cable lead-in between the antenna and TV.

If necessary, you should consult your dealer or an experienced radio/television technician for additional suggestions. You may find helpful the following booklet, prepared by the Federal Communications Commission:

"How to Identify and Resolve Radio-TV Interference Problems"

This booklet is available from the U.S. Government Printing Office, Washington, DC 20402, stock number 004-000-00345-4.

Table of Contents

Preface

This manual is a supplement to the *Apple Dot Matrix Printer User's Manual Part I*; it contains information specific to the Apple II computer. You should read the *Apple Dot Matrix Printer User's Manual Part I* first, and then turn to this manual only when directed to do so.

This manual is arranged so that you can find the information you want quickly and easily, without having to read things you aren't interested in just yet. Here is an overview of what this manual contains:

- Chapter 1, "Connecting Your Printer and Computer," explains how to hook your printer and computer together.
- Chapter 2, "Printing With Standard Programs," describes how to use your printer with some of the programs commonly available for the Apple II.
- Chapter 3, "Printing With Standard Languages," discusses the use of the Apple Dot Matrix Printer with the standard Apple II programming languages.
- Chapter 4, "Utility Program HEXCODE," contains a utility program to help you get the most out of your printer.

The following chart show which chapters are recommended for various types of readers. You may fall into more than one of the categories (for example, a first-time user who sets up his or her own system).

Reader	Chapters 1	2	3	4
The person who sets up the system	•			
First-time user; wants to use ready-made programs		•		
Experienced Apple computer user		•	•	
Programmer			•	•
Business person		•		

Warning: This Equipment Is Intended To Be Electrically Grounded.

This product is equipped with a three-wire grounding type plug, a plug having a third (grounding) pin. This plug will only fit into a grounding-type AC outlet. This is a safety feature. If you are unable to insert the plug into the outlet, contact a licensed electrician to replace the outlet and, if necessary, install a grounding conductor. Do not defeat the purpose of the grounding-type plug.

Connecting Your Printer and Computer

Making the Connection

To operate your printer with an Apple II computer, you need a Parallel Interface Card (Apple Product #670-0021) and a connecting cable (Apple Product #590-0042B).

Follow the instructions that came packed with the Parallel Interface Card; they tell you how to install the card inside your Apple computer and how to run the cable from the card out one of the slots in the back panel of the Apple II.

Warning
Make sure the printer is turned **off** before you perform the hook-up.

The Parallel Interface Card contains a set of small switches that can be used to tailor its output to the requirements of different printers. For use with your Apple Dot Matrix Printer, select the "standard" switch settings—switches 1, 2, 3, 6 and 7 off; 4 and 5 on.

Normally the Parallel Interface Card plugs into slot 1 of your Apple II Computer. Included with it is a cable a few inches long that runs from the card to the back of your computer where it terminates in a new connector. The connecting cable to your Apple Dot Matrix Printer plugs into this new connector.

The other end of the connecting cable has a metal connector on it that plugs into the interface socket on the back of the Apple Dot Matrix Printer. Look at the end of the connector; notice that one edge is wider than the other. The wide edge faces up. Gently push the connector into the socket, and then secure it by bringing the two wire loops toward each other. They snap into notches on both ends of the connector and prevent it from coming loose.

Connecting the Apple Dot Matrix Printer to the Apple II requires no special software. The printer is activated simply by addressing the peripheral slot number into which the Parallel Interface Card is plugged. The procedures for doing so are discussed in Chapter 3 of this supplement, "Printing With Standard Languages."

What to Do Next

Return to your *Apple Dot Matrix Printer User's Manual Part I* and finish reading Chapters 2 and 3, "Setting Up Your Printer" and "Care and Handling." Then, if you plan to use your printer with already-written programs, such as Apple Writer, read Chapter 2 in this supplement, "Printing With Standard Programs." If you wish to learn how to use your printer directly from programming languages such as BASIC or Pascal, read Chapter 3 in this supplement, "Printing With Standard Languages," and Chapters 4 and 5 in your *Apple Dot Matrix Printer User's Manual Part I*, "Controlling Your Printer" and "Advanced Control Codes."

Printing With Standard Programs

The easiest way to operate your Apple Dot Matrix Printer is through the standard programs that are available for use with your computer. This approach offers a simple and foolproof way to create written records without having to understand any programming languages. However, it does not let you take advantage of certain features of the printer that are accessible only if you write your own programs.

This chapter uses Apple Writer II—one of the more popular packaged programs for the Apple II—as an example of how to use your Apple Dot Matrix Printer with a standard program. Using your printer with other programs is a similar process. This chapter assumes that you are able to operate your Apple computer, but it does not require that you be able to write programs.

Your printer must be hooked up to your computer according to the directions in the preceding chapter.

Apple Writer II

Apple Writer II is a word processing program that you can use to create, edit, and print documents. Apple Writer II offers many sophisticated features for composing and editing text. When used with the Apple Dot Matrix Printer, Apple Writer II can print in a variety of formats. For complete information, see the *Apple Writer II* manual.

Using Control Codes

Control codes: Special character sequences used to control your printer from your Apple computer. Chapters 4 and 5, "Controlling Your Printer" and "Advanced Control Codes," in your *Apple Dot Matrix Printer User's Manual Part I* describe control codes.

You can include printer control codes in your text files to select a combination of character and line pitches, tab locations, margins, and so on. You can also place underlining and boldface commands at the appropriate places in the text. However, you cannot simply enter the control codes as you would all other text.

Many standard programs for creating text files are designed to react to some of the characters used in control codes; they treat them as immediate commands instead of letting them pass through to the printer. For example, if you press (ESC) while creating a file with the Pascal Editor program, you will terminate the program. You cannot normally place the Escape character in the file you are editing.

There is only one way to get around this problem: use the HEXCODE program described in chapter 4 of this supplement. The utility program, HEXCODE, allows you to embed control codes by using their hexadecimal equivalents, and then processing the file.

For Experts Only: There is yet another way to include control codes in files. You can use the CHR or CHR$ functions (described in the next chapter) to create a file of control code characters such as Escape and Tab, and then copy the characters into any file in which you want them.

Printing With Standard Languages

You can use the same commands to send text to your Apple Dot Matrix Printer that you use to save text in a file on a disk or display it on the video monitor. However, the commands used are somewhat different for each programming language. The discussions in this chapter are intended to provide an introduction to using your Apple Dot Matrix Printer with these languages. For further details, refer to the sections in your programming manuals that cover output devices such as printers.

The Printer Is a Device

Device: A piece of equipment attached to a computer. The printer, video monitor, keyboard, and disk drives are all devices.

Each disk drive is one of a number of **devices** that provide input and output to the computer. The Apple Dot Matrix Printer is another such device. It responds when you call it by name (usually PRINTER:). Of course, it only records data, but in its role as a recorder you can address it just as you would a disk drive.

Apple II Monitor Program

The Apple II Monitor program provides direct access to the Apple II's memory for reading and modifying the 6502 machine code. You enter the Monitor from BASIC by executing the statement CALL -151.

Your Apple Dot Matrix Printer can print listings of portions of memory. Simply precede each Monitor command by the number of the slot that the Parallel Interface Card occupies, followed by CONTROL-P. This will send the output from that command to the printer. After the command has been executed, output will return to the video monitor.

For example, to print a listing of the contents of memory from locations addr1 to addr2 (where addr1 and addr2 are hexadecimal addresses), enter the following after the asterisk Monitor prompt:

① CONTROL-P addr1.addr2

(Assuming that the Parallel Interface Card is in slot 1).

For further information about Monitor commands, see the reference manual for your Apple II.

For Experts Only: If you're writing a program in machine language and want to send characters to your Dot Matrix Printer or to the video monitor, send the output (in ASCII) character by character to $FD18. Each character must have the most significant bit set.

You must specifiy the device that you are sending characters to by loading the accumulator with the device's slot number and executing a JSR to $FD95. If you're using DOS, specify the slot number with a PR# statement.

If you have ROM Applesoft, you can use this short-cut:

```
300: LDA #$20
302: LDY #$03
304: JSR $DB3A
307: RTS

320: 48 45 4C 4C 4F 00
     H  E  L  L  O
```

Integer BASIC and Applesoft BASIC

Integer BASIC is the simplest BASIC language available for your Apple II computer. Applesoft BASIC is an extended form of Integer BASIC with additional commands and improved capabilities.

When either language is running, accessing your Apple Dot Matrix Printer is simply a matter of executing a PR# statement to select the slot that the Parallel Interface Card occupies. Normally this is slot 1, so the statement is

```
PR#1
```

Deferred statement: A statement that is part of a program and that is executed only when the program runs.

Immediate statement: A statement that is not part of a program and that is executed immediately after you press RETURN to end the statement.

This statement can be executed in either **deferred** or **immediate** mode. After it has been executed, all subsequent output from the computer will go to the printer. The statement `PR#0` will return output to the video monitor.

Remember: If you are using Integer BASIC or Applesoft loaded from the Disk Operating System (DOS), all deferred PR# statements must be prefaced by (CONTROL)-(D). See your DOS manual.

To print listings of Integer BASIC and Applesoft programs, you use DOS. Here's how:

1. Place the DOS SYSTEM MASTER disk that came with your disk drive in your first drive and turn on your computer to start the system.
2. After the drive stops running and the DOS title appears on the video display, remove the DOS SYSTEM MASTER disk and replace it with the disk containing the file to be printed (for example, MYPROG).
3. Type the following lines, ending each by pressing (RETURN):

```
LOAD MYPROG
PR#1
LIST
```

By the Way: Step 3 assumes that the Parallel Interface Card is in slot 1 of your Apple II. If it is in another slot, type the appropriate number after PR#.

4. The Apple II retrieves MYPROG from the disk and sends its contents to the printer. If you type RUN instead of LIST in the last line, MYPROG executes, and any resulting output is sent to the printer.
5. To stop sending data to the printer, type PR#0.

By the Way: Applesoft BASIC always sets the eighth bit of its output characters, even when executing CHR$ commands. DIP switch 2-6, therefore, must be kept closed. See Chapter 4, "Controlling Your Printer," in your *Apple Dot Matrix Printer User's Manual Part I* for a description of DIP switches.

Tabbing

Your printer's Parallel Interface Card is programmed to make formatted listings of BASIC programs. Unfortunately, this feature interferes with the Tab function. So, instead of

```
PRINT "HELLO"; TAB(70); "THERE!"
```

use

```
PRINT "HELLO"; : POKE 36,70 : PRINT "THERE!"
```

The rest of this section on tabbing presents more complex information and is meant more for the expert only. If this doesn't apply to you, skip down to the next section, "Initializing With Poke Statements."

Instead of using POKE 36,X to tab, you can use the following machine language routine. You must have DOS, and you must use Call statements in place of PR# statements. Enter the Monitor by typing

```
CALL -151
```

First determine the value for the three variable parameters (X,Y, and Z) for the slot with the Parallel Interface Card:

Slot:	1	2	3	4	5	6	7
X	C1	C2	C3	C4	C5	C6	C7
Y	F9	FA	FB	FC	FD	FE	FF
Z	01	02	03	04	05	06	07

Type the following machine language routine, using the values given above for X, Y, and Z:

```
3B0: A9 Z
   : 20 95 FE
   : A9 8D
   : 20 ED FD
   : A9 C5
   : 85 36
   : A9 03
   : 85 37
   : 4C EA 03
   : 20 02 X
   : 48
   : AD  Y 07
   : 85 24
   : 68
   : 60
```

To check your typing, enter

```
3B0L
```

and compare your listing to the one below for a Parallel Interface Card in slot 1:

```
03B0- A9 01     LDA #$01
03B2- 20 95 FE  JSR $FE95
03B5- A9 8D     LDA #$8D
03B7- 20 ED FD  JSR $FDED
03BA- A9 C5     LDA #$C5
03BC- 85 36     STA $36
03BE- A9 03     LDA #$03
03C0- 85 37     STA $37
03C2- 4C EA 03  JMP $03EA
03C5- 20 02 C1  JSR $C102
03C8- 48        PHA
03C9- AD F9 07  LDA $07F9
03CC- 85 24     STA $24
03CE- 68        PLA
03CF- 60        RTS
```

Return to BASIC by typing

```
3D0G
```

Save the program in a file with

```
BSAVE TABBER, A$3B0, L$20
```

The first time you want to use TABBER, load it by typing

```
BLOAD TABBER
```

The first time you need the printer, you must initialize the Parallel Interface Card's slot by typing

```
CALL 944
```

Subsequently, you don't need to reinitialize the card each time you want to print something. Use

```
CALL 954
```

Warning
Do not use PR# statements to initialize the printer; use CALL 944 and CALL 954.

If you use any Poke statements or control characters to modify the Parallel Interface Card's operation, issue them after CALL 944; you don't need to repeat them after each CALL 954.

Here is a sample program that illustrates how to use TABBER:

```
100  D$ = CHR$(4)
110  PRINT D$;"BLOAD TABBER"
120  CALL 944
130  PRINT"THIS WILL BE PRINTED BY THE PRINTER
140  PRINT D$;"PR#0"
150  PRINT"AND NOW BACK TO THE SCREEN"
160  CALL 954
170  PRINT"NOW FOR A TABBING DEMO"
180  FOR J = 1 TO 76 STEP 5
190  PRINT TAB (J); J;
200  NEXT J
210  PRINT
220  PRINT D$;"PR#0"
230  END
```

Initializing With Poke Statements

Neither the PR# or IN# statements initialize the Parallel Interface Card. This can cause problems if you want to modify the operating parameters of the Parallel Interface Card, because you need to send a character through the card before poking in the new parameters. This section provides you with Poke statements to use in place of PR# and IN# statements.

The normal way to reset the video display and keyboard is

```
D$ = CHR$(4)
PRINT D$;"PR#0"
PRINT D$;"IN#0"
```

However, this works only after a Print statement, and is ignored after a Get or Print statement that ends with a comma or semicolon. To avoid having to use the extra Print statement, use

```
CALL -375 : REM THIS IS IN#0
CALL -365 : REM THIS IS PR#0
CALL 1002 : REM THIS RECONNECTS DOS
```

Use CALL 1002 if you need DOS while the Parallel Interface Card is enabled. However, if speed is of the essence, don't use CALL 1002 until the information has been transferred to the printer, since DOS does slow data input and output.

Apple II Pascal

With Apple II Pascal, you can use the File Transfer command to print the contents of Pascal text files. Just follow this procedure:

1. Rlace your Apple1 disk in your first disk drive (volume #4:) and turn on the computer. The message `Welcome Apple1` should come up on your video monitor.
2. Press Ⓕ and then Ⓣ to activate the Transfer routine in the Filer program. Make sure that the printer is on and selected.
3. Place the disk containing the file to be printed (let's call it MYTEXT) in the first disk drive.
4. When the program asks `Transfer?` type the complete name of the file to be printed—for example, `#4:MYTEXT.TEXT`.
5. When it asks `To where?` type `PRINTER:`.
6. The printer will now print the contents of the file MYTEXT.

Pascal provides a variety of commands for sending data to devices such as your Apple Dot Matrix Printer. They are discussed in detail in your *Apple Pascal Language Reference Manual*. The first step is to assign a file variable to the printer by means of the Rewrite procedure. For example, if the file variable were called ADMP, the first procedure for using the printer would be

```
REWRITE (ADMP, 'PRINTER:')
```

All subsequent procedures using the printer would then refer to it as ADMP; for instance,

```
WRITE (ADMP, 'h')
```

would print a lowercase *h*. When the program has finished sending data to the printer, use the statement

```
CLOSE (ADMP)
```

Using Control Codes

Control codes: Special character sequences used to control your printer from your Apple computer. Chapters 4 and 5, "Controlling Your Printer" and "Advanced Control Codes," in your *Apple Dot Matrix Printer User's Manual Part I* describe control codes.

Just as including control codes in text files is not simple (the process is described in the section, "Using Control Codes," in Chapter 2 of this supplement), including control codes in program files is similarly somewhat difficult. There are two ways to get around the problem; use the CHR or CHR$ functions, or use the HEXCODE program described in Chapter 4 of this supplement.

Methods for using the CHR and CHR$ functions are given in Chapter 4, "Controlling Your Printer," in your *Apple Dot Matrix Printer User's Manual Part I*. This technique is not available in Integer BASIC, which lacks a CHR$ function.

The utility program HEXCODE allows you to enter control codes by using their hexadecimal equivalents, and then processing the program file. (See Chapter 4 of this supplement.)

Utility Program HEXCODE

To use all of the capabilities of the Apple Dot Matrix Printer, you must be able to send it virtually any eight-bit code. However, this occasionally turns out to be difficult: either the character you want cannot be entered directly from the keyboard, or its occurrence in the program you are using produces undesirable side effects. For example, if you are using the Pascal Editor, you will encounter problems when you try to send control codes to your Apple Dot Matrix Printer. The reason is that the Pascal Editor responds to many of the same characters, such as Escape and Tab, that are in printer control codes. It interprets them as immediate editing commands and refuses to put them into your file, where they could be transferred to the printer.

Perhaps the easiest way to handle control codes is to enter them as their hexadecimal equivalents. The utility program HEXCODE does this.

Both a BASIC and a Pascal listing for HEXCODE are given in this chapter. You do not need an expert's knowledge of programming to use this program. If you follow the instructions carefully, you will be able to add it to your collection of computer tools.

By using HEXCODE, you can substitute a two-digit hexadecimal value for any troublesome character. Thus you can create all 256 of the eight-bit ASCII characters. You can use the single HEXCODE program in two different ways:

- to send characters directly from the keyboard to the printer
- to send the contents of a text file to the printer (a text file can be created using Applewriter II).

When used in any of these manners, HEXCODE performs a translation function on all characters sent or transferred. Most text goes through unchanged; but whenever a dollar sign ($) occurs in the input, the two characters immediately following it are interpreted as a hexadecimal number. The dollar sign and two-digit number are replaced by the equivalent single ASCII character, which appears at that point in the output.

For an explanation of hexadecimal numbers and a list of the hex codes for all ASCII characters, see Appendix C of your *Apple Dot Matrix Printer User's Manual Part I.*

For example, to place an Escape character in text, you enter a dollar sign and its hex code—$1B. In this form, you can edit and print your program without trouble. When you send or transfer the text via HEXCODE, the sequence $1B is automatically replaced by the ASCII Escape character. If your file is a program, you must process your program through HEXCODE before you compile it. HEXCODE creates a new file in which all the hex values are replaced by control characters.

Creating HEXCODE in BASIC

To add HEXCODE to your library of BASIC utility programs for use with the Apple Dot Matrix Printer, follow these steps:

1. Remove any existing program in memory by typing `NEW`. Carefully enter the following listing. The REM statements are explanatory only; they are not part of the program and may be omitted.

By the Way: If you want to use a character other than $ to flag hex number entries in the text being processed by HEXCODE, just change the HEXFLAG assignment in the second line of the program.

2. Run the program to see if it works. Correct any syntax errors you have made.
3. When you have successfully corrected any errors, save the program in a file on a disk.

Now whenever you want to process text with HEXCODE, just use the Run statement to execute the code file you saved.

BASIC HEXCODE Listing

```
10    REM  PROGRAM HEXCODE.  TAKES INPUT FROM EITHER THE KEYBOARD OR A
           TEXT FILE AND OUTPUTS TO THE PRINTER. IF THE CHARACTER
           IS A "$", THEN THE NEXT TWO CHARACTERS FOLLOWING THE $ ARE
           CONVERTED TO HEXADECIMAL AND OUTPUT TO THE PRINTER AS A
           SINGLE VALUE.

15    REM INITIALIZE VARIABLES
20    HXFLAG$ = "$"
30    HEX$ = "123456789ABCDEF":D$ =  CHR$ (4)

35    REM DISPLAY PROMPTS AND GET INFORMATION FROM USER
40    TEXT : HOME : PRINT "*** CONVERT HEXIDECIMAL TO ASCII ***"
50    PRINT : PRINT : PRINT : PRINT "INPUT (FILENAME OR K": INPUT "       FOR
       KEYBOARD): ";A$
60    PRINT : PRINT : PRINT "OUTPUT (FILENAME OR P": INPUT "       FOR
       PRINTER): ";B$
70    IF B$ = "P" THEN OUTFILE$ = "PRINTER"
80    IF A$ < > "K" THEN INFILE$ = A$: GOTO 90
85    INFILE$ = "KEYBOARD"
90    PRINT : PRINT : INPUT "DISCARD THE 8TH BIT? (Y/N) ";A$
100   IF A$ = "Y" THEN BIT8% = 128: GOTO 120
110   BIT8% = 0

115   REM SETUP ERROR DETECTION FOR END OF FILE
120   ONERR  GOTO 500
125   REM GET THE INPUT FROM THE FILE OR FROM THE KEYBOARD
130   IF INFILE$ < > "KEYBOARD" THEN  GOSUB 700
140   INPUT A$
143   B$="0":REM SET B$ = "0" IN CASE RETURN IS PRESSED WITH NO
                VALUE IN A$.
145   REM CHECK EACH CHARACTER IN THE INPUT STRING: A$
150   DIGIT% = 0
160   FOR PLACE = 1 TO LEN (A$)
170   IF ASC(MID$ (A$,PLACE,1)) > 127 THEN B$ = CHR$( ASC ( MID$
      (A$,PLACE,1)) - BIT8%): GOTO 190
180   B$ = MID$ (A$,PLACE,1)
190   IF B$ = HXFLAG$ THEN DIGIT% = 2: GOTO 210
200   GOSUB 300:DIGIT% = DIGIT% - 1
210   NEXT PLACE

215   REM CHECK TO SEE IF THE OUTPUT IS TO THE PRINTER OR TO THE CONSOLE
220   IF OUTFILE$ = "PRINTER" THEN PR# 1: PRINT : PR# 0
230   IF OUTFILE$ < > "PRINTER" THEN  PRINT
240   GOTO 140

300   REM IF DIGIT% IS 2 OR 1, THEN CONVERT THE NEXT TWO CHARACTERS TO
          ITS HEXADECIMAL VALUE AND SEND IT TO THE OUTFILE$.
305   IF DIGIT% = 2 THEN  GOSUB 600:HEXVLUE = VLUE% * 16
310   IF DIGIT% = 1 THEN  GOSUB 600:HEXVLUE = VLUE% + HEXVLUE: IF OUTFILE$ =
      "PRINTER" THEN PR# 1: PRINT  CHR$ (HEXVLUE);: PR# 0
320   IF DIGIT% = 1 AND OUTFILE% < > "PRINTER" THEN PRINT  CHR$ (HEXVLUE);
330   IF (DIGIT% < > 1 AND DIGIT% < > 2) AND OUTFILE$ = "PRINTER" THEN PR# 1:
      PRINT B$;: PR# 0
340   IF (DIGIT% < > 1 AND DIGIT% < > 2) AND OUTFILE$ < > "PRINTER" THEN
      PRINT  CHR$ B$;
350   RETURN

500   REM ERROR CHECKING FOR END OF FILE (EC=5). IF IT'S NOT THE END OF FILE
      (EOF) THEN DISPLAY THE ERROR MESSAGE
505   EC =  PEEK (222)
510   IF EC = 5 THEN  PRINT D$;"CLOSE";INFILE$: END
520   POKE 216,0
530   RESUME

600   REM CONVERT THE CHARACTER IN B$ TO THE HEXADECIMAL EQUIVALENT
603   VLUE% = 0
605   FOR COUNT = 1 TO LEN (HEX$)
610   IF B$ = MID$ (HEX$,COUNT,1) THEN VLUE% = COUNT
620   NEXT COUNT
630   RETURN

700   REM OPEN INFILE$ AND SET TO READ MODE
705   PRINT D$;"OPEN";INFILE$
710   PRINT D$;"READ";INFILE$
720   RETURN
```

Creating HEXCODE in Pascal

To add HEXCODE to your library of Pascal utility programs for use with the Apple Dot Matrix Printer, follow these steps:

1. Using the Pascal Editor, carefully enter the following into the SYSTEM.WRK file. Remember that in Pascal syntax, spaces and line endings are not important, but all of the punctuation must be accurate. The comments within curly brackets { } are explanatory only; they are not part of the program and may be omitted.

By the Way: If you want to use a character other than $ to flag hex number entries in the text being processed by HEXCODE, just change the HEXFLAG assignment in the second line of the program.

2. From the Pascal main command line, press C to compile the text you just entered. The Compiler will inform you of any syntax errors you have committed; correct them before continuing.
3. When you have successfully compiled HEXCODE, save your text and code files on a disk.

Now whenever you want to process text with HEXCODE, just use the X command on the Pascal main command line to execute the codefile you saved.

Pascal HEXCODE Listing

```
PROGRAM HEXCODE;

CONST   HEXFLAG = '$';                          {Establish flag character}
        HEX = '123456789ABCDEF';              {Define hexadecimal numerals}

VAR     INFILE, OUTFILE : TEXT;
        BUFF1, BUFF2 : STRING;
        PLACE, DIGIT, VALUE, BIT8 : INTEGER;
        KBD : BOOLEAN;

BEGIN
KBD := FALSE;                                  {Initialize keyboard input flag}
                                                 {Display title and prompts}
WRITELN ('*** CONVERT HEXADECIMAL TO ASCII ***', CHR(13));
WRITE ('Input filename (or K for Keyboard): ');
READLN (BUFF1);
WRITE ('Output Filename (or P for Printer): ');
READLN (BUFF2);
IF (BUFF2 = 'P') OR (BUFF2 = 'p') THEN BUFF2 := 'PRINTER:';
REWRITE (OUTFILE, BUFF2);                    {Open output file or printer}
WRITE (CHR(13), 'Discard 8th bit? (Y for yes): ');
READLN (BUFF2);
WRITELN;
IF (BUFF2='Y') OR (BUFF2='y') THEN BIT8 := 128 ELSE BIT8 := 0;
                {Use built-in file INPUT for keyboard, or open input file}
IF (BUFF1 = 'K') OR (BUFF1 = 'k') THEN KBD := TRUE ELSE
RESET (INFILE, BUFF1);
BUFF2 := '0';                  {Establish 1-character length for BUFF2}
{When using keyboard, exit from loop with Control-C}
WHILE (KBD AND NOT EOF) OR (NOT EOF(INFILE)) DO
  BEGIN                       {Fetch string from keyboard or input file}
  IF KBD THEN READLN (BUFF1) ELSE READLN (INFILE, BUFF1);
  DIGIT := 0; {Reset digit counter}
  {Examine characters in input string}
  FOR PLACE := 1 TO LENGTH(BUFF1) DO
    BEGIN
    IF ORD (BUFF1[PLACE]) > 127 THEN BUFF2[1] :=
      CHR(ORD(BUFF1[PLACE])-BIT8) ELSE BUFF2[1] :=
      BUFF1[PLACE];                  {Strip out 8th bit if requested}
    IF BUFF2[1] = HEXFLAG THEN DIGIT := 3 ELSE
                  {If it's the hex flag, set digit counter and go on}
      BEGIN       {Next after hex flag is most significant hex digit}
      IF DIGIT = 2 THEN VALUE := POS(BUFF2, HEX) * 16;
                        {Following is least significant hex digit}
      IF DIGIT = 1 THEN
        BEGIN
        VALUE := VALUE + POS(BUFF2, HEX);
        WRITE (OUTFILE, CHR(VALUE))
        END;
      IF DIGIT < 1 THEN WRITE (OUTFILE, BUFF2[1])
                  {If not following the hex flag, just type it plain}
      END;
    DIGIT := DIGIT-1                          {Decrement digit counter}
    END;
  WRITELN (OUTFILE)
  END;
CLOSE (INFILE);                                {Close input and output}
CLOSE (OUTFILE, LOCK)
END.
```

Executing HEXCODE in BASIC or Pascal

When you execute HEXCODE, a message appears on the video monitor that asks you to identify the input by entering either a filename or pressing K for *keyboard*. If the text to be processed is in a file, enter its full pathname. If you want to enter text from the keyboard, just press K. In both cases, end your entry by pressing RETURN.

The next video monitor message asks for the output filename. If you want the processed result saved in a file, enter its full filename. If the file already exists, HEXCODE will write over it without informing you. If you want the processed result sent directly to the Apple Dot Matrix Printer, just press P or . In any case, end your entry by pressing RETURN.

If there is any problem finding or opening the file or files you have specified, your Apple computer will stop execution of HEXCODE and display an error message on the video monitor.

Finally, HEXCODE will ask whether you want the eighth (most significant) bit of all input characters discarded (in other words, set to zero in the output regardless of its value in the input). To discard the eighth bit, press Y and then press RETURN; if you make any other entry, the eighth bit will be retained unchanged.

After this entry, HEXCODE will either perform the file operation you requested or wait for you to enter text from the keyboard (if you pressed K for input). If your input is from the keyboard, you can edit each line on the video monitor before sending it to HEXCODE. When you finish entering text from the keyboard, press CONTROL-C.

By the Way: If you need to place the hex flag character in your input as an ordinary text character, not as the start of a hex number, you can enter its hex equivalent. For example, the input $24 becomes translated to plain $ in the output (24 is the hex code for the character $).

When writing Pascal programs, you may want to place "illegal" ASCII characters between quotation marks in Write statements. Such characters (Escape, Return, Tab, and so on) will of course produce unwanted actions when you try to print a listing of your program. However, you can write the hex codes for such characters in your original source file, and then use HEXCODE to prepare a second source file that contains the "illegal" characters in their natural form. This second file can then be used as a source for compiling p-code (Pascal), while the first file remains as text that can be listed and edited.

20525 Mariani Avenue
Cupertino, CA 95014
(408) 996-1010
TLX 171576

030-0739-A